Capacity and Expressiveness of Genomic Tandem Duplication

Siddharth Jain
Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
sidjain@caltech.edu

Farzad Farnoud (Hassanzadeh)
Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
farnoud@caltech.edu

Jehoshua Bruck
Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
bruck@caltech.edu

***Abstract*—The majority of the human genome consists of repeated sequences. An important type of repeats common in the human genome are tandem repeats, where identical copies appear next to each other. For example, in the sequence $AGTC\underline{TGTG}C$, $TGTG$ is a tandem repeat, namely, generated from $AGTCTGC$ by a tandem duplication of length 2. In this work, we investigate the possibility of generating a large number of sequences from a small initial string (called the seed) by tandem duplications of bounded length. Our results include *exact capacity* values for certain tandem duplication string systems with alphabet sizes $2, 3$, and 4. In addition, motivated by the role of DNA sequences in expressing proteins via RNA and the genetic code, we define the notion of the *expressiveness* of a tandem duplication system, as the feasibility of expressing arbitrary substrings. We then *completely* characterize the expressiveness of tandem duplication systems for general alphabet sizes and duplication lengths. Noticing that a system with capacity = 1 is expressive, we prove that for an alphabet size ≥ 4, the capacity is strictly smaller than 1, independent of the seed and the duplication lengths. The proof of this limit on the capacity (note that the genomic alphabet size is 4), is related to an interesting result by Axel Thue from 1906 which states that there exist arbitrary length sequences with no tandem repeats (square-free) for alphabet size ≥ 3. Finally, our results illustrate that duplication lengths play a more significant role than the seed in generating a large number of sequences for these systems.**

***Index Terms*—Expressiveness, tandem repeats, finite automata, square-free strings.**

I. Introduction

More than 50% of the human genome consists of repeated sequences [6]. These repeats are mainly of two types i) interspersed repeats and ii) tandem repeats. Interspersed repeats are caused by transposons. A transposon (jumping gene) is a segment of DNA that can copy or cut and paste itself into new positions of the genome. Tandem repeats are thought to be caused by slipped–strand mispairings [10]. Slipped–strand mispairings are thought to occur when one DNA duplex becomes misaligned with the other.

Tandem Repeats are common in both prokaryote and eukaryote genomes. They are not only present in intergenic regions but also in both coding and non-coding regions. They are thought to be the cause of several genetic disorders. The effects of tandem repeats on several biological processes is understood by these disorders. They can result in generation of toxic or malfunctioning proteins, chromosome fragility, expansion diseases, silencing of genes, modulation of transcription and translation [12] and rapid morphological changes [4].

A process that leads to tandem repeats is *tandem duplication* which allows substrings of certain lengths to be duplicated next to their original position. For example, from the sequence $AGTCGTCGCT$, a tandem duplication of length 2 can give $AGTCGT\underline{CGCG}CT$, which if followed by a duplication of length 3 can give $AGTCG\underline{TCGTCG}CGCT$. The significance of sequences with tandem repeats and the fact that much of our unique DNA was likely originally a repeated sequence motivates us to study the *capacity* and *expressiveness* of *string systems* with tandem duplication. The model of a string duplication system consists of a starting string (seed) of finite length, a set of duplication rules, and the set of all sequences that can be obtained by applying the duplication rules to the seed a finite number of times. The notion of capacity, introduced in [3], represents the average number of m-ary bits per symbol that are required asymptotically to encode a sequence in the string system, where m is the alphabet size. The notion of expressiveness defined formally later answers the question whether each of the finite length sequences for a given alphabet can be obtained as a substring of some sequence in the string system. Expressiveness and capacity are closely related. More precisely, it is not difficult to show *if we have a system that is not expressive then capacity* < 1 [9].

Tandem duplications have already been studied in [1], [2], [7], [8]. However the main concern of these works is to determine the place of tandem duplication rules in the Chomsky hierarchy of formal languages. A study related to our work can be found in [3] where the authors show that for a fixed duplication length the capacity is 0 in a tandem duplication string system. Further, they find a lower bound on the capacity of these systems, when duplications of all length are allowed. In this paper, we consider tandem duplication string systems, where we restrict the maximum size of the block being tandemly duplicated to a certain *finite* length. In the rest of the paper, the term tandem duplication string system refers to this kind of string duplication system.

Example 1: To illustrate the notion of expressiveness and capacity for tandem duplication string systems, consider a string system on binary alphabet where the seed is 01 and the maximum allowable block size for duplication is 2. It is easy to check that the set of strings that can be generated by this system start with a 0 and end with a 1. In fact, it can be proved that all binary strings of length n which start with 0 and end with 1 can be generated by this system. The proof is based on the fact that every n length string which starts with 0 and ends with 1 can be rewritten as $0^{r_1}1^{r_2}........0^{r_{m-1}}1^{r_m}$, where each $r_i \geq 1$ and m is even. Hence a natural way to generate such a string from seed $=$ 01 is to duplicate 01 $\frac{m}{2}$ times and then duplicate each 0 or 1 at position i, r_i times.

Expressiveness: 11010 cannot be generated by this system. However, it can be generated as a substring of 0110101 in the following way:

$$01 \rightarrow 0101 \rightarrow 010101 \rightarrow 0\underline{11010}1.$$

If every binary string can be generated as a substring of some larger string in the duplication system, then we say that the system is expressive. In this case, since every binary string starting with 0 and ending with 1 can be generated, we can generate every binary string as a substring. Hence, the system is expressive.

Capacity: The number of n-length strings in this string system is 2^{n-2} and therefore the capacity is 1 bit/symbol.

Observing these facts for an alphabet of size 2, one can ask related questions on expressiveness and capacity for higher alphabet sizes and duplication lengths. However, counting the number of n- length sequences for capacity calculation and characterizing expressive systems for higher alphabets is non-trivial. In this paper, we study these questions and develop tools to answer them. It is interesting to observe that the string system over binary alphabet in the above example can be represented by the finite automata given in Figure 1. The regular expression for the language defined by the finite automata is given below which exactly represents all binary strings that start with 0 and end with 1.

$$R_{01} = (0^+1^+)^+ \qquad (1)$$

A. Summary of Results

Given a finite automata, one can use Perron-Frobenius theory [5], [9] to count the number of sequences which can be generated by a finite automata. In this paper, we use finite automata as a tool to calculate capacity for string duplication systems with tandem repeats over higher alphabet. In our results, we find the exact capacity for a tandem duplication string system over ternary alphabet with seed 012 and duplication size atmost 3 to be 0.876036. Furthermore, we show that no expressive tandem duplication string system over ternary alphabet with maximum duplication length 3, exists. However, if the maximum duplication length is 4 and the seed is 012, then we get an expressive system. This shows that for such string duplication systems, the maximum duplication length plays a more significant role in generating a larger number of strings than the seed. Further, to emphasize this fact we state a result from [3] that over all tandem duplication string systems with a given alphabet size and maximum duplication length, an expressive tandem duplication string system has maximum capacity. We also find that for alphabet size > 3, an expressive tandem duplication string system does not exist which shows that full capacity (i.e. capacity = 1) cannot be achieved by a tandem duplication string system for alphabet size > 3.

It is easy to check that for a binary alphabet, any sequence of length ≥ 4 has a tandem repeat. The dependence of expressiveness and capacity on alphabet size is intuitively connected to a result by Thue [11] which states that for an alphabet of size > 2, there exists a square-free sequence (sequence with no tandem repeat) of every length. In our proofs of results on expressiveness, we elaborate on this connection with Thue's result. The rest of the paper is organized as follows. In section II, we give the preliminary definitions and notation. In section III, we provide our results on capacity and expressiveness. We conclude the paper in section IV.

II. Preliminaries

Let Σ be some finite alphabet. An n-string $x = x_1x_2...x_n \in \Sigma^n$ is a finite sequence where $x_i \in \Sigma$ and $|x| = n$. The set of all finite strings over alphabet Σ is denoted by Σ^*. For two strings $x \in \Sigma^n$ and $y \in \Sigma^m$, their concatenation is denoted by $xy \in \Sigma^{n+m}$. For a positive integer m and a string s, s^m denotes the concatenation of m copies of s. A string $v \in \Sigma^*$ is a substring of x if $x = uvw$, where $u, w \in \Sigma^*$.

A string system $S \subseteq \Sigma^*$ is represented as a tuple $S = (\Sigma, s, \mathcal{T})$, where $s \in \Sigma^*$ is a finite length string called seed, which is used to start the duplication process, and $\mathcal{T}$ is the duplication rule [3].

Tandem Duplication of length k: $\mathcal{T}^{tan}_k : \Sigma^* \rightarrow \Sigma^*$, is defined as

$$\mathcal{T}^{tan}_k(x) = uvvw, \ where \ x = uvw, \ |v| = k. \qquad (2)$$

Furthermore, let $\mathcal{T}^{tan}_{\leq k}$ denote the set of tandem duplications of length at most k, i.e., $\mathcal{T}^{tan}_{\leq k} = \{\mathcal{T}^{tan}_{k'} | k' \leq k\}$. With this notation, for system considered in Example 1, $S = (\{0,1\}, 01, \mathcal{T}^{tan}_{\leq 2})$.

We denote by $N_S(n)$ the number of strings in S of length n. The *capacity* of the string system S is defined as:

$$cap(S) = \limsup_{n \to \infty} \frac{\log_{|\Sigma|} N_S(n)}{n}. \qquad (3)$$

Next, we define the notion of *expressiveness*. A string system S is *expressive* if for each $y \in \Sigma^*$, there exists a $z \in S$, such that y is a substring of z.

III. Results and proofs

Our first result is on the capacity of a tandem duplication string system over a ternary alphabet.

Theorem 1: For a tandem duplication string system $S = (\{0,1,2\}, 012, \mathcal{T}^{tan}_{\leq 3})$, $cap(S) = 0.876036$.

Proof: We prove this theorem by showing that S is represented by the finite automata given in Figure 2. The finite automata is designed in such a way so that it covers tandem duplications of length 1, 2 and 3. The self loops on each state

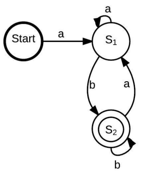

Fig. 1: Finite Automata for $S = (\{a, b\}, ab, \mathcal{T}^{tan}_{\le k})$, where $k \ge 2$ (In binary alphabet, we do not gain anything by increasing k above 2), a and b are distinct symbols.

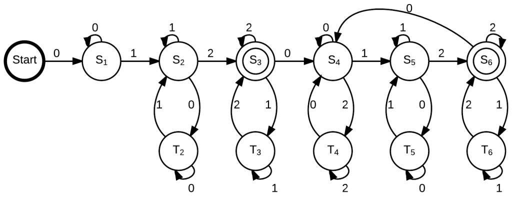

Fig. 2: Finite Automata for $S = (\{0, 1, 2\}, 012, \mathcal{T}^{tan}_{\le 3})$.

cover duplication of length 1, connections in between state pairs S_i and T_i cover duplications of length 2, and the edge between states S_6 and S_4 covers duplications of length 3. In the rest of the proof, we show that the finite automata we construct in this way indeed represents S under consideration.

The regular expression for the language defined by this finite automata is

$$R_{012} = (0^+1^+)^+2^+(1^+2^+)^*[0^+(2^+0^+)^*1^+(0^+1^+)^*2^+(1^+2^+)^*]^* \tag{4}$$

Let $L_{R_{012}}$ denote the language defined by the regular expression R_{012} or equivalently finite automata in Figure 2. We claim that

Claim 1: $L_{R_{012}} \subseteq S$

Before moving to the proof of Claim 1, we define the *de-duplication* process. Consider a de-duplication map $\mathcal{D}_{\le k} : \Sigma^* \to P^{\Sigma^*}_{\le k}$. Here $P^{\Sigma^*}_{\le k}$ is the power set of strings in Σ^* which do not have a tandem repeat $\alpha\alpha$, where $|\alpha| \le k$. For $x \in \Sigma^*$, $\mathcal{D}_{\le k}(x)$ is the set of strings in $P^{\Sigma^*}_{\le k}$ from which x can be obtained by tandem duplications of size at most k. For example, $\mathcal{D}_{\le 3}(010100001) = \{01\}$, $\mathcal{D}_{\le 3}(2122221212) = \{212\}$, $\mathcal{D}_{\le 4}(012101212) = \{012, 0121012\}$.

Now, we show that for every string $x \in L_{R_{012}}$, $012 \in \mathcal{D}_{\le 3}(x)$ or in other words every string $x \in L_{R_{012}}$ can be de-duplicated to 012 using $\mathcal{D}_{\le 3}$.

The regular expression R_{012} in (4) can be represented as $R_{012} = B_1{B_2}^*$, where

$$B_1 = (0^+1^+)^+2^+(1^+2^+)^* \tag{5}$$

$$B_2 = 0^+(2^+0^+)^*1^+(0^+1^+)^*2^+(1^+2^+)^* \tag{6}$$

It is easy to check that de-duplication $\mathcal{D}_{\le 3}$ converts $a^+ \to a$, $a^* \to \epsilon$ or a, $(ab)^+ \to ab$, $(ab)^* \to \epsilon$ or ab, $(abc)^+ \to abc$ and $(abc)^* \to \epsilon$ or abc, where a, b and c are distinct.

Next, we show that applying de-duplication $\mathcal{D}_{\le 3}$ on B_1 gives 012 and on B_2 gives either 02012 or 012.

i) De-duplication $\mathcal{D}_{\le 3}$ on B_1: de-duplication

$$(0^+1^+)^+2^+(1^+2^+)^* \to 012(12)^* \to 012.$$

ii) De-duplication $\mathcal{D}_{\le 3}$ on B_2 :

$$0^+(2^+0^+)^*1^+(0^+1^+)^*2^+(1^+2^+)^* \to 0(20)^*1(01)^*2(12)^*$$

$$\to 0(20)^*1(01)^*2 \to 0(20)^*12 \to 02012 \text{ } or \text{ } 012.$$

Therefore, $B_1{B_2}^*$ can be de-duplicated to 012 by applying $\mathcal{D}_{\le 3}$ since

$$B_1{B_2}^* \to 012(02012)^* \to 012$$
$$or$$
$$B_1{B_2}^* \to 012(012)^* \to 012.$$

Hence, any $x \in L_{R_{012}}$ can be de-duplicated to 012 by $\mathcal{D}_{\le 3}$ or in other words, each $x \in L_{R_{012}}$ can be obtained by tandem duplications of length atmost 3 if the seed $s = 012$. Therefore, $L_{R_{012}} \subseteq S$.

Now, we claim that every $x \in S$ also belongs to $L_{R_{012}}$, i.e.

Claim 2: $S \subseteq L_{R_{012}}$

To prove this we need to show two things for the finite automata in Figure 2:

i) It can generate 012.

ii) If the automaton can generate pqr, with $p, q, r \in \Sigma^*$ and $|q| \le 3$, it can also generate pq^2r.

(i) holds trivially (See the path $Start - S_1 - S_2 - S_3$ in Figure 2).To prove ii) we look at the adjacency matrix of the finite automata and show that for each state C all the 1, 2 and 3 length paths that end in C, we have a corresponding path with the same label which starts in C and ends in some state which is equivalent to C. Due to space limitations, the proof details are omitted here.

After proving $S = L_{R_{012}}$, we use Perron-Frobenius Theory [5], [9] to count the number of sequences which can be generated from this deterministic finite automata. We calculate the maximum absolute eigen value e^* of the adjacency matrix B of the strongly connected component of the finite automata in Figure 2 (i.e., $S_4, S_5, S_6, T_4, T_5, T_6$). B is given by

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

for B, $e^* = 2.618034$. By Perron-Frobenius Theory, $cap(S) = \log_3 e^* = 0.876036$ (upto 6 decimal places). ■

Intuition as to why the capacity is less than 1: If we observe the regular expression for the finite automata, we see that we cannot generate a string which has 210, 021 or 102 as a substring (which also means that the system is not expressive). This further puts constraints on strings of $size > 3$ that can be generated using this finite automata. Hence, we cannot achieve full capacity.

We know from Example 1 that the capacity for $S = (\{0, 1\}, 01, \mathcal{T}^{tan}_{\le 2})$ is 1. Using this result, we can calculate the capacity for another tandem duplication string system given by $S_1 = (\{0, 1, 2\}, 012, \mathcal{T}^{tan}_{\le 2})$. One way to see this is noting that S_1 can be decoupled into two string systems namely $(\{0, 1\}, 01, \mathcal{T}^{tan}_{\le 2})$ and $(\{1, 2\}, 12, \mathcal{T}^{tan}_{\le 2})$. Similarly, we can calculate the capacity for $S_2 = (\{0, 1, 2, 3\}, 0123, \mathcal{T}^{tan}_{\le 3})$ since it can also be decoupled in a similar manner to string systems of lower alphabet size and maximum duplication length 3. We omit the proof details here. Our capacity results are listed in Table I.

$alphabet(\Sigma)$	$seed(s)$	k	Capacity
$\{0,1\}$	01	1	$= 0$
$\{0,1\}$	01	≥ 2	$= 1$
$\{0,1,2\}$	012	2	$= 0.630930$
$\{0,1,2\}$	012	3	$= 0.876036$
$\{0,1,2,3\}$	0123	3	$= 0.694242$

TABLE I: Capacity values for different string systems with starting string s that allow tandem duplications upto size k.

The next few theorems are on the expressiveness of tandem duplication string system,

Theorem 2: The system $S = (\{0,1,2\}, s, \mathcal{T}^{tan}_{\leq 3})$, where s is any arbitrary string $\in \{0,1,2\}^*$, is not expressive.

Proof: Before, we move to the proof, let us define the notion of a *3-irreducible* string. A string $x \in \{0,1,2\}^*$ is 3-irreducible if it does not have a tandem repeat $\alpha\alpha$, such that $|\alpha| \leq 3$. For example, 01201, 01210, 02101, 01210121 are 3-irreducible strings. 01212, 021021, 01112 are not 3-irreducible. To prove Theorem 2, we construct a 3-irreducible string which is not a substring of any $y \in S$.

At any stage of duplication in S, we can either do a tandem duplication of length 1 or 2 or 3. The string z on which the duplication is to be performed can be represented in the following way $z = uvw$, where $|v| \leq 3$ and v is the string that is to be tandemly duplicated. From tandem duplication of v in z, we get $z^* = uvvw$. We consider the following 3 cases and observe the 3-irreducible substrings in z^* which do not possibly appear in z:

Case 1: $|v| = 1$, $v = a_1$.
Here $z = ua_1v$ and $z^* = ua_1a_1v$, the new substrings that we see in z^* are not 3-irreducible. Since, they have a repeat a_1a_1.

Case 2: $|v| = 2$, $v = a_1a_2$.
Here $z = ua_1a_2v$, and $z^* = ua_1a_2a_1a_2v$, the new possible 3-irreducible substrings that we see in z^* of length ≥ 3 have either $a_1a_2a_1$ as suffix or $a_2a_1a_2$ as prefix, which means that if any new 3-irreducible substring is generated in this step, either i) the letter on its first and third position is same or ii) the letter on its last and third last position are same.

Case 3: $|v| = 3$, $v = a_1a_2a_3$.
Here $z = ua_1a_2a_3v$, and $z^* = ua_1a_2a_3a_1a_2a_3v$, the new possible 3-irreducible substrings that we see in z^* of length ≥ 4 have either $a_1a_2a_3a_1$ or $a_1a_2a_3a_1a_2$ as suffix or $a_3a_1a_2a_3$ or $a_2a_3a_1a_2a_3$ as prefix, which means that if any new 3-irreducible substring is generated in this step, either i) the letter on its first and fourth position is same or ii) the letter on its last and fourth last position are same.

Consider, an arbitrary 3-irreducible string $\in \{0,1,2\}^*$ of length ≥ 4. Let $b_1b_2b_3b_4$ be its prefix and $c_4c_3c_2c_1$ be its suffix. From the 3 cases considered above, we have the following conditions, one of which has to be satisfied by the 3-irreducible substrings that can be generated by S,
$b_1 = b_3$ *or* $b_1 = b_4$ *or* $c_1 = c_3$ *or* $c_1 = c_4$.
Now, we need to show that there are 3-irreducible strings that do not satisfy any of the above 4 conditions. Consider 3-irreducible strings of the form $t = (cbab)^m cba$ or $(abcb)^m a$, $m \geq 1$ and a, b and c are distinct symbols $\in \{0,1,2\}$. The 4-length suffix for strings of this form is $bcba$ and the 4-length prefix is either $abcb$ or $cbab$. None of these suffix or prefix satisfies any of the four conditions listed above. Hence, if not present in the seed s, 3-irreducible substrings of this type cannot be generated by S. Since the seed s is of finite length, we have for some m, an 3-irreducible string t with length $> |s|$ which cannot be generated as a substring of some string in S. Hence, S is not expressive. ■

Theorem 3: The system $S = (\Sigma, s, \mathcal{T}^{tan}_{\leq k})$, where $|\Sigma| \geq 4$, s is any arbitrary seed $\in \Sigma^*$ and k is some finite natural number, is not expressive, which also implies $cap(S) < 1$.

Proof: We can extend and imitate the idea used in the proof of Theorem 2 (this time we have k cases). Consider an arbitrary square-free string $\in \Sigma^*$ of length $\geq k+1$. Let $b_1b_2.......b_kb_{k+1}$ be its prefix and $c_{k+1}c_k.......c_1$ be its suffix. After considering k cases, we will get the following $2k-2$ conditions, one of which has to be satisfied by any square-free string y that is a substring of some $t \in S$.
$b_1 = b_{1+i}$ for some $i \in \{2,3,4,...,k\}$ *or* $c_1 = c_{1+j}$ for some $j \in \{2,3,4,...,k\}$.
Now, we show a construction of an irreducible string $\in \Sigma^*$ which does not satisfy any of the above listed conditions. Let $\Sigma = \{e_1, e_2, e_3, ..., e_{|\Sigma|}\}$. Let $G = \{x : x \in \{e_2, ..., e_{|\Sigma|}\}^*, |x| \geq k-1, x \text{ is square-free}\}$. Then, for any $y \in G$, it is easy to check that $t = e_1ye_1$ does not satisfy any of the $2k-2$ conditions listed above. By Thue [11], for alphabet size ≥ 3, for any length there exists a square-free string. Therefore, for each length $m \geq k-1$, there exists a $y \in G$ with $|y| = m$. Since the seed s is of finite length, for some m we have an irreducible string t with length $> |s|$ which cannot be generated as a substring of some string in S. ■

Theorem 4 : The system $S = (\{0,1,2\}, 012, \mathcal{T}^{tan}_{\leq 4})$ is an expressive string system.

Proof: Theorem 4 can be proved by an induction argument on the length of the substring that we want to generate. The system S considered in Theorem 4 clearly generates all the strings which can be generated using the system considered in Theorem 1. Looking at the finite automata in Figure 2 or R_{012} in Eq. (4) for string system considered in Theorem 1, it is easy to check that all possible strings of lengths 1 and 2 over ternary alphabet can be obtained as substrings. Hence, all substrings of length 1 and 2 can be obtained using S considered in Theorem 4. Now to prove that all substrings of length 3 can also be obtained using S, it will be sufficient to prove that 021, 102 and 210 can be obtained as substrings using S, since other substrings of length 3 can be obtained by the system considered in Theorem 1 (again by observing the finite automata in Figure 2 or R_{012} in (4)).

To generate $210, 021$ and 102 as substrings, here is the method:

$$012 \rightarrow 0\underline{1212} \rightarrow \underline{01\mathbf{210}121}2$$

$$012 \rightarrow \underline{012012} \rightarrow 01\underline{202}\underline{0}12 \rightarrow 0\underline{12\mathbf{021}202}012$$

$$012 \rightarrow \underline{012012} \rightarrow 01\underline{2020}12 \rightarrow 012\underline{020\mathbf{102}01}2$$

For a substring of length 4, we have the following 3 cases: In case 1 and case 2 below, w is assumed to have at least one occurence of each letter in the alphabet.

Case 1: The first 3 letters in the substring w are all distinct, i.e., if $w = w_1w_2w_3w_4$, then $w_1 \neq w_2 \neq w_3 \neq w_1$. For generating such w as a substring, we first generate $w_1w_2w_3$ and then do a tandem duplication of w_3 if $w_4 = w_3$, of w_2w_3 if $w_4 = w_2$ and of $w_1w_2w_3$ if $w_4 = w_1$.

Case 2: Exactly 2 letters in the first 3 letters of w are same, i.e., if $w = w_1w_2w_3w_4$, then either $w_1 = w_2 \neq w_3$, or $w_1 = w_3 \neq w_2$, or $w_1 \neq w_2 = w_3$. If $w_1 = w_2$, then we first generate $w_1w_3w_4$ and then do a tandem duplication of w_1 to get $w = w_1w_1w_3w_4$. If $w_1 \neq w_2$, then we first generate $w' = w_4w_1w_2w_3$ as a substring, and then do a tandem duplication of w' to get w. (Note: w' is of type considered in *Case 1* since w_4 is different from both w_1 and w_2) .

Case 3: w has ≤ 2 distinct letters, such a w has a tandem repeat. Therefore if $w = xyyz$, where either $|y| = 2$ and $|x| = |z| = 0$, or $|y| = 1$ and $|x| \leq 1$, $|z| = 2 - |x|$, then we first generate xyz and do a tandem duplication of y to get w.

Until now, we have shown that all substrings w of length ≤ 4 can be generated.

For generating a substring w with $|w| > 4$, we use inductive argument. Assuming all substrings of length $\leq m$ can be generated (here $m \geq 4$), we need to prove that we can generate all substrings of length $m + 1$. Consider an arbitrary $w = a_1a_2....a_ma_{m+1}$. By induction assumption $w' = a_1a_2...a_m$ can be generated. Here, we have two cases: i) If all the three letters in the alphabet occur atleast once in $a_{m-3}a_{m-2}a_{m-1}a_m$, then w can be generated as a substring by a tandem duplication of some suffix of size ≤ 4 of w'. ii) If atleast one letter in the alphabet does not occur in $a_{m-3}a_{m-2}a_{m-1}a_m$, then $a_{m-3}a_{m-2}a_{m-1}a_m$ is a sequence over binary alphabet and hence is of the form $xyyz$ ($|y| = 1$ or 2), therefore w can be generated as a substring by tandem duplication of y on $a_1.....a_{m-4}xyza_{m+1}$. (Note $|a_1....a_{m-4}xyza_{m+1}| \leq m$). Hence, we have proved Theorem 4. ■

Remark 1: It is important to note that in case (ii) above the binary sequence $a_{m-3}a_{m-2}a_{m-1}a_m$ has a tandem repeat. If the original alphabet size $|\Sigma|$ was > 3, then this is not guaranteed over $|\Sigma| - 1$-ary sequence of any length because of Thue's result [11].

Remark 2: For $S = (\{0, 1\}, s, \mathcal{T}^{tan}_{\leq 1})$, $(01)^m$ cannot be generated as a substring of any string $\in S$ for some m.

Table II gives a complete characterization of the expressiveness of tandem duplication string systems.

Theorem 5 [3]: For an expressive tandem duplication system $S = (\Sigma, s, \mathcal{T}^{tan}_{\leq k})$, $cap(S) \geq cap(S')$, where $S' = (\Sigma, s', \mathcal{T}^{tan}_{\leq k})$. i.e. the capacity cannot be improved by only changing the seed if a tandem duplication string system is expressive for some seed s.

Remark 3: By Theorem 3, $|\Sigma| \leq 3$ in Theorem 5.

$alphabet(\Sigma)$	$seed(s)$	k	Expressiveness	Reason
$\{0\}$	0	≥ 1	Yes	Trivial
$\{0, 1\}$	arbitrary	1	No	Remark 2
$\{0, 1\}$	01	≥ 2	Yes	Example 1
$\{0, 1, 2\}$	arbitrary	≤ 3	No	Theorem 2
$\{0, 1, 2\}$	012	≥ 4	Yes	Theorem 4
Size ≥ 4	arbitrary	arbitrary	No	Theorem 3

TABLE II: Expressiveness of tandem duplication string systems where the maximum duplication length is k.

IV. Conclusion

It is proved in [8] that the language defined by any tandem duplication string system with alphabet size ≥ 3 and maximum duplication length ≥ 4 is not regular if the seed has abc as a substring where a, b and c are distinct symbols. In this paper, we show a regular language construction when maximum duplication length is 3. Using the method of decoupling that we use to calculate capacities for $S = (\{0, 1, 2\}, 012, \mathcal{T}^{tan}_{\leq 2})$ and $S = (\{0, 1, 2, 3\}, 0123, \mathcal{T}^{tan}_{\leq 3})$ in this paper, we can also show that if the maximum duplication length is 3, then the language defined by the tandem duplication string system is regular irrespective of the seed and the alphabet size. Due to space limitations, we have omitted the proof here.

V. Acknowledgements

This work was supported in part by the NSF Expeditions in Computing Program - The Molecular Programming Project. The first author would also like to thank Gaurav Sinha for some helpful discussions.

References

[1] J. Dassow, V. Mitrana, and G. Paun, "On the regularity of duplication closure," *Bulletin of the EATCS*, vol. 69, pp. 133-136, 1999.

[2] J. Dassow, V. Mitrana, and A. Salomaa, "Operations and language generating devices suggested by the genome evolution," *Theoretical Computer Science*, vol. 270, no.1 , pp. 701-738, 2002.

[3] F. Farnoud, M. Schwartz, and J. Bruck, "The Capacity of String-Replication Systems, " http://arxiv.org/pdf/1401.4634.pdf

[4] J. W. Fondon and H. R. Garner,"Molecular origins of rapid and continuous morphological evolution," *Proceedings of the National Academy of Sciences*, vol. 101, no. 52, pp. 18 058 – 18 063, 2004.

[5] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.

[6] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860-921, 2001.

[7] P. Leupold, C. Martin-Vide, and V. Mitrana, "Uniformly bounded duplication languages," *Discrete Applied Mathematics*, vol. 146, no. 3, pp. 301-310, 2005.

[8] P. Leupold, V. Mitrana, and J. M. Sempere, "Formal languages arising from gene repeated duplication," in *Aspects of Molecular Computing*, Springer, 2004, pp. 297-308.

[9] D. Lind and B. H. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1985.

[10] N. Mundy and A. J. Helbig, "Origin and evolution of tandem repeats in the mitochondrial DNA control region of shrikes (lanius spp.)," *Journal of Molecular Evolution*, vol. 59, no. 2, pp. 250-257, 2004.

[11] A. Thue, " über unendliche Zeichenreihen," *Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl., Cristiana* 7, 1906.

[12] K. Usdin, "The biological effects of simple tandem repeats: lessons from the repeat expansion diseases," *Genome research*, vol. 18, no. 7, pp. 1011-1019, 2008.